

**REMARKS****Status of the Claims**

Claims 1, 3-8, 10-14, and 16-20 are currently present in the Application, and claims 1, 8, and 14 are independent claims. Claims 1, 3-4, 6-8, 10-11, 13-14, and 16-20 have been amended, claims 2, 9, and 15 have been cancelled, and no claims have been added.

**Examiner Interview**

Applicants note with appreciation the telephonic interview conducted between Applicants' representative and the Examiner on March 13, 2007. During the telephonic interview, the Examiner and Applicants' representative discussed the 102 reference (Richard M Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1). In particular, Applicants' representative discussed that Applicants' invention selects a processor type for each source code subtask included in a source code file, and independently generates object code for each of the source code subtasks to execute on the selected processor types. In contrast, Stallman generates an object file for a processor type using an entire source code file, and does not teach or suggest individually generating object code for each source code subtask. Applicants' representative suggested incorporating limitations included in dependent claim 2 into Applicants' independent claims, as well as further describing independent object code generation steps for a particular processor type for each source code subtask. Such amendments are included in this response. In addition, claims 14-20 have been amended to remove "means for" language in the claim limitations. No agreement was reached regarding the claims.

**Drawings**

Applicants note with appreciation the Examiner's acceptance of Applicants' formal drawings filed concurrently with the application.

**Claim Rejections - Alleged Anticipation Under 35 U.S.C. § 102**

Claims 1-4 and 7 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Stallman (Richard M Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1). Applicants respectfully traverse these rejections.

Applicants have incorporated limitations previously found in original claim 2 into Applicants' independent claim 1. As such, Applicants have canceled claim 2 in this response and, therefore the rejection to this claim is moot. Applicants have also amended claim 1 to further define individual processor-type selection and object code generation for each source code subtask. Support for such amendments can be found in Applicants' specification on page 52, line 8 through page 55, line 28. Therefore, no new matter is added with such amendments. As amended, independent claim 1 is directed to a method for compiling source code for a plurality of heterogeneous processors with limitations comprising:

- receiving source code that includes a plurality of source code subtasks;
- independently selecting a processor type from the plurality of heterogeneous processor types for each of the plurality of source code subtasks, the independent selection comprising:
  - selecting a first processor type from the plurality of heterogeneous processor types for a first source code subtask included in the source code; and
  - selecting a second processor type from the plurality of heterogeneous processor types for a second source code subtask included in the source code, wherein the second processor type is different than the first processor type; and
- creating an object file that includes a first object code corresponding to the first source code subtask and a second object code corresponding to the second source code subtask, wherein the first object code is adapted to be processed by the first processor type and the second object code is adapted to be processed by the second processor type.

Applicants individually retrieve source code subtasks included in a source code file, select a particular processor type for each subtask, and create a processor-type specific object code for each of the subtasks. In contrast, Stallman generates object code for a processor type for an **entire** source code file. Stallman states:

“Compilation can involve up to four stages: preprocessing, compilation proper, assembly, and linking, always in that order. The first three stages **apply to an individual source file, and end by producing an object file**; linking combines all the object files (those newly compiled, and those specified as input) into an executable file.” (page 15, section 3.2, emphasis added)

As can be seen from the above excerpt, as well as the options disclosed in Stallman’s Section 3.2, Stallman treats a source code file as a single entity and compiles a source code file into a single object file. As a result, Stallman never teaches or suggests *“selecting a first processor ...for a first source code subtask included in the source code, selecting a second processor type for a second source code subtask included in the source code, and creating an object file that includes a first object code corresponding to the first source code subtask and a second object code corresponding to the second source code subtask, wherein the first object code is adapted to be processed by the first processor type and the second object code is adapted to be processed by the second processor type”* as claimed by Applicants.

Therefore, since Stallman does not teach or suggests all the limitations included in claim 1 as amended, amended claim 1 is allowable over Stallman.

Notwithstanding the fact that claim 3 is dependent upon claim 1 and, therefore, allowable for at least the same reasons as claim 1 discussed above, claim 3 adds limitations to claim 1 of:

- wherein the selection of the first processor type is performed during compilation, the method further comprising:
  - retrieving the first source code subtask from the plurality of source code subtasks;

- determining whether the first source code subtask includes a program directive corresponding to one of the plurality of processors; and
- performing the selection of the first processor type in response to the determination.

Applicants “*determine whether the first source code subtask includes a program directive corresponding to one of the plurality of processors.*” Meaning, the program directive is included in the first source code subtask. The Office Action uses an excerpt to reject Applicants’ determination limitation but, after further review, Stallman does not teach such limitation. Rather, Stallman teaches that the **compilers** themselves have program directives that can be changed by a user. Stallman states:

“Earlier we discussed the standard option “-b” which chooses among different installed compilers for completely different target machines...In addition, each of these target machines types can have its own special options...Some configurations of the compiler also support additionally special options...These options are defined by the macro TARGET\_SWITCHES in the machine description. The default for the options is also defined by that macro, which enables you to change the defaults.” (page 75, section 3.17, emphasis added)

As can be seen from the above excerpt, Stallman teaches the compilers having program directives for a particular processor type, but never teaches or suggests “*determining whether the **first source code subtask includes a program directive** corresponding to one of the plurality of processors; and performing the selection of the first processor type in response to the determination*” as claimed by Applicants.

Therefore, since Stallman does not teach or suggests all the limitations included in claim 3, claim 3 is allowable over Stallman. Each of the remaining claims 4 and 7 depend upon claim 1 and, therefore, are allowable for at least the same reasons that claim 1 is allowable as discussed above.

### **Claim Rejections – Alleged Obviousness Under 35 U.S.C. § 103**

Claim 5 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman. Applicants respectfully traverse these rejections. Claim 5 depends upon

claim 1 and, therefore, is allowable for at least the same reasons that claim 1 is allowable as discussed above.

Claims 6 and 8-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman in view of Ansari et al. (U.S. Patent No. 6,473,897, hereinafter "Ansari"). Applicants respectfully traverse these rejections. Applicants have canceled claims 9 and 15 in this response and, therefore, rejections to these claims are moot.

Claim 6 depends upon claim 1 and, therefore, is allowable for at least the same reasons that claim 1 is allowable as discussed above.

Claim 8 is an information handling claim including similar limitations as claim 1. The Office Action does not suggest that Ansari teaches or suggests the limitations included in claim 1, and indeed Ansari does not teach such limitations. Therefore, claim 8 is allowable over Stallman in view of Ansari for at least the same reasons that claim 1 is allowable over Stallman as discussed above.

Claim 14 is computer program product claim including similar limitations as claim 1. The Office Action does not suggest that Ansari teaches or suggests the limitations included in claim 1, and indeed Ansari does not teach such limitations. Therefore, claim 14 is allowable over Stallman in view of Ansari for at least the same reasons that claim 1 is allowable over Stallman as discussed above.

Each of the remaining claims 10-13 and 16-20 depend, either directly or indirectly, upon one of the allowable independent claims 8 or 14. Therefore, each of claims 10-13 and 16-20 are allowable for at least the same reasons as their respective independent claims are allowable as discussed above.

### **Conclusion**

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/

Leslie A. Van Leeuwen, Reg. No. 42,196

Van Leeuwen & Van Leeuwen

Attorney for Applicants

Telephone: (512) 301-6738

Facsimile: (512) 301-6742